**Felipe Canales**

**UBID: 50106507**

# Is machine learning useful in the identification of encrypted malware Packet Capture network traffic?

## 1. Introduction

Cyber-attacks are becoming more complex and occur at a fast pace, requiring the adoption of machine learning (ML) methods for detection at the speed of a machine. The typical use of machine learning in a network environment necessitates the selection of relevant characteristics from available data by subject matter experts in order to identify harmful network traffic. The features retrieved from network traffic are next subjected to pre-processing (also known as feature engineering), which includes the use of different mathematical methods to prepare the data for use by the machine learning model. Malware, often known as malicious software, is a type of computer program that is designed to infiltrate and harm or disable computer systems without the user's knowledge or permission [1]. In order to obtain commands from an attacker, these malicious programs communicate with a command and control (C&C) server via which they are distributed. Over 90% of small-to-medium-sized businesses (SMBs) have seen a rise in the number of malware detections, according to Malwarebytes, a prominent cybersecurity solution. In fact, some firms have seen a 500 percent increase in malware detections in only one month, in March 2017 [2]. Network traffic information can be used to detect malware in real-time, which has the ability to prevent—-or at the very least drastically reduce—-malware propagation on a network. In recent years, there has been much study into the use of machine learning for the detection of malicious network traffic; this is especially attractive when the data is encrypted since conventional pattern-matching methods cannot be utilized. In order to illustrate and better understand the impact that these errors have on popular machine learning algorithms, researchers have designed and carried out experiments that demonstrate how typical algorithms perform when faced with real-world data from social media networks. By analyzing the experimental findings, we are able to determine the circumstances in which particular classes of algorithms fail on the job of identifying encrypted malware traffic categorization and provide specific suggestions for practitioners in light of the real-world limitations that have been identified.

## 2. Related Works

When it comes to network administration, traffic categorization is essential for tasks such as flow priority and load balancing. In light of the fast growth in network applications and network traffic (as well as the accompanying increase in malicious traffic), there is now a pressing need to distinguish between benign and malicious data as soon as possible. Techniques for traffic categorization may be divided into three main groups [3]. However, although port-based methods depend on apps that use standard ports, there is nothing that stops an application from utilizing non-standard ports instead. Anonymity and privacy are often ensured via the use of port switching and port obfuscation. Using a second method, you can look for signatures in the payload, which may be useful when dealing with dynamic port changes but is less effective when dealing with encrypted data. Flow statistics, such as flow rate, packet length, inter-packet time, and other parameters inferred from flows are employed in the third class of methods; for example, this technique is becoming more popular, especially when deep packet inspection (DPI) is not feasible, either because the data is encrypted or because the flow rates are very high [4]. However, the use of software-defined protocols and new applications adds to the difficulty of feature-based traffic categorization. In the previous study, you can find a thorough review of feature-based traffic classification algorithms.

Sophisticated machine learning methods have developed during the past decade, with the benefit that they can intuitively learn characteristics from the data. This obviates feature engineering, which is problematic with quickly developing applications. Many of these ML methods utilize header data, or a mix of header data and raw data (approaches utilizing both headers and raw data are frequently termed multimodal [5]. Many publications utilize private data that makes it impossible to repeat tests or compare against the performance of new methods. In contrast, we utilize just raw data and base our assessments on the publicly accessible UNSW-NB15 dataset [6]. The concept of utilizing just raw data has been addressed in a few publications, which we discuss next.

A common strategy has been to represent non-image data as pictures in order to leverage the massive amount of research in deep learning (DL) for image processing applications. This technique was used to visualize graphs related to social networks [7]. A similar technique was used in related work14 to classify network traffic. 784 bytes were chosen here, either from the TCP session payload or from the payloads of all levels. The reasoning for this was that the first few hundred bytes would include connection data. These 784 bytes were transformed to 28 by 28 grayscale pictures (as specified by MNIST) and then trained using a 2D CNN architecture inspired by LeNet-5. The authors evaluated their own dataset (USTC-TFC2016, which is currently accessible on Github). The authors next explored the usage of 1D-CNNs to classify traffic into various application categories using the raw byte stream (again, 784 = 282 bytes as input) [5]. In a supplemental study16, the first 900 bytes are utilized and represented as 30 x 30 grey-scale pictures for their Deep-Full-Range classifier, which is composed of three parallel structures — a stacked auto-encoder, a two-layer 1D-CNN with local normalization, and an LSTM-based classifier. The process for choosing a suitable classifier for detecting malicious communications during operation is not entirely apparent.

In the "DeepPacket" framework, a more complex DL architecture comprised of a stacked auto-encoder and a CNN was used to distinguish encrypted from unencrypted traffic and VPN traffic from non-VPN traffic [8]. They used the first 1480 bytes of the IP payload as input (padded if necessary), as their research of the ISCX VPN-nonVPN dataset indicated that 96 percent of packets have a payload length of less than 1480. Additionally, investigators18 suggest the use of attention-based LSTMs in conjunction with a hierarchical attention network (HAN) to classify encrypted communication into many classes. The input here consisted of ten packets, each of which had been truncated (or padded) to 1500 bytes. Additionally, another malware detection approach dubbed DeepMAL has recently proposed. It employs a combination of 1D CNN and LSTM and uses the first 1024 bytes of the payload as input, based on an evaluation of the packet lengths of benign and malicious data [9].

Methods for malicious network communication in the past have relied on either port-based categorization or deep packet inspection and signature matching techniques to communicate. When using port-based methods, it is assumed that applications always use well-known port numbers that have been registered by the Internet Assigned Numbers Authority (IANA) [7] and that the application uses well-known port numbers that have been registered by the Internet Assigned Numbers Authority (IANA). Network intrusion detection systems (NIDS) and limiting firewalls, according to Marín,  Casas, and  Capdehourat [9], are able to identify malicious programs by using non-standard ports to avoid detection. Even well-known apps such as Skype make use of dynamic port numbers in order to avoid being blocked by restrictive firewalls [10]. Madhukar and Williamson in [11] shown that port-based categorization incorrectly classifies network flow traffic 30-70 percent of the time, according to their findings.

By inspecting payload contents and utilizing conventional pattern matching or signature-based methods, Etienne in [12] was able to identify malicious data by employing deep packet inspection to detect malicious traffic. Etienne utilized Snort [12], an intrusion detection program, to identify malicious traffic by comparing the contents of packets with signatures or strings that were generated by the application. On top of that, Snort additionally offers a popular Intrusion Protection System (IPS) rule set that is updated by the community [14]. However, just around 1% of the ruleset is TLS specific, demonstrating that conventional pattern matching methods are not often employed for TLS based malware. When categorizing Peer-to-Peer (P2P) traffic. Yoon et al. [13] show that deep packet inspection may decrease false positive and false negative rates by 5 percent when using deep packet inspection. Michael et al. reported in [15] that they were able to identify network programs with 100 percent accuracy by examining the full packet content. The main drawbacks of these techniques are the violation of user privacy as well as the enormous cost associated with decrypting and analyzing each individual packet.

BotFinder, a network-flow information-based method for detecting bot infestations, was introduced by Tegeler and colleagues in [16]. To detect abnormalities in the network activity between two endpoints, the system employs traces, which are a series of chronologically ordered flows. Other network information, such as the average time interval, the average duration, the average amount of bytes sent and received at the source and destination locations, and so on, were utilized as features in a local shrinkage-based clustering method [17]. In [18], Prasse et al. developed a neural network-based malware detection system that took into account network flow characteristics such as port value, connection length, number of bytes transmitted and received, time interval between packets, and domain name characteristics. We no longer utilize domain name features or DomainName System (DNS) data as features as a result of the introduction of DNS over TLS, which encrypts both the DNS data and the domain name system data using TLS. In [20], Loko and colleagues published a k-NN-based classification method that may be used to detect servers that were accessed by malware through HTTPS traffic.

According to Anderson and McGrew in [21], a novel method that analyzes network flow information and applies supervised machine learning techniques to detect encrypted malware traffic has been presented. For the purpose of collecting and training the machine learning algorithm on innocuous network data, they set up a demilitarized zone (DMZ). A DMZ is a sub-network that is used to segregate services that are accessible from the outside world from internal systems. Services that are externally linked are those that connect to the internet in order to offer a variety of services. Because it was based on supervised learning models, it produced findings that were straightforward to understand [21]. The machine learning model aided in the high-speed processing of network data as well as the ability to make real-time forecasts. [23] It also made use of regularization, which is an essential component of training, to pick the characteristics that were the most discriminating. Due to the fact that the DMZ separates such services and is only utilized in commercial organizations, the network traffic data gathered by them is not a true reflection of the whole amount of traffic on the internet. The findings may not be applicable to ordinary internet users such as students or home users, as stated in [21], due to the fact that this data is solely representative of corporate users, i.e., those who work in commercial companies.

## 3. Conclusion

In conjunction with a rise in global use of HTTPS and advancements in malware detection methods, we expect to see an increase in the number of malware samples that use HTTPS and encryption to avoid detection and conceal their harmful activities. Concerns have been raised because encryption may create problems with conventional detection methods. A significant challenge is identifying such risks in a manner that is practical, quick, and does not jeopardize the security of the users. In recent years, machine learning techniques have shown their ability to transcend conventional constraints and have been used in the training of models on malware network traffic. Afterward, these models may be used to identify similar malicious network activity and flag a machine as being infected with malware. Furthermore, the system may be isolated in order to avoid the spread of malware on the internal network in the future.

The main reason for this study is the difficult issue of categorizing encrypted network traffic as harmful or benign without the use of decryption or deep packet inspection, which was the core focus of this research. The findings demonstrate that XGBoost outperformed the other algorithms and achieved the greatest accuracy of 99.15 percent. The findings also support the notion that machine learning models may be utilized to address the multi-class issue in the first place. As a result, we may infer that encrypted malware network traffic is different from regular network traffic and that it varies from one malware family to another. Successful identification of an infected host, as well as the specification of the malware family with which the host that is infected, may be accomplished using this feature.

There is much room for improvement in future work. The next stage would be to gather more data for training and testing the models, as well as to identify any new characteristics that may be helpful for categorization.

## References

[1] B. Arslan, S. Gunduz, and S. Sagiroglu, "A review on mobile threats and machine learning-based detection approach," in *2016 4th International Symposium on Digital Forensic and Security (ISDFS)*, 2016.

[2] A. S. Shekhawat, F. D. Troia, and M. Stamp, "Feature analysis of encrypted malicious traffic," *Expert Syst. Appl.*, vol. 125, pp. 130–141, 2019.

[3] S.-H. Yoon, J.-W. Park, J.-S. Park, Y.-S. Oh, and M.-S. Kim, "Internet application traffic classification using fixed IP-port," in *Management Enabling the Future Internet for Changing Business and New Computing Services*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 21–30.

[4] P. Wang, X. Chen, F. Ye, and Z. Sun, "A survey of techniques for mobile service encrypted traffic classification using deep learning," *IEEE Access*, vol. 7, pp. 54024–54033, 2019.

[5] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "DISTILLER: Encrypted traffic classification via multimodal multitask deep learning," *J. Netw. Comput. Appl.*, vol. 183–184, no. 102985, p. 102985, 2021.

[6] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW- NB15 network data set," 2015.

[7] K. Hegde, M. Magdon-Ismail, R. Ramanathan, and B. Thapa, "Network signatures from image representation of adjacency matrices: Deep/transfer learning for subgraph classification," *arXiv [cs. CV]*, 2018.

[8] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Siberian, "Deep packet: a novel approach for encrypted traffic classification using deep learning," *Soft Comput.*, vol. 24, no. 3, pp. 1999–2012, 2020.

[9] G. Marín, P. Casas, and G. Capdehourat, "DeepMAL -- deep learning models for malware traffic detection and classification," *arXiv [cs. CR]*, 2020.

[10] S. Masood, M. A. Shahid, M. Sharif, and M. Yasmin, "Comparative analysis of peer-to-peer networks," *International Journal of Advanced Networking and Applications*, vol. 9, no. 4, pp. 3477–3491, 2018.

[11] A. Madhukar and C. L. Williamson, "'A longitudinal study of P2P traffic classification,'" 2006, pp. 179–188.

[12] L. Etienne, "Malicious traffic detection in local networks with a snort," Available: https://infoscience.epfl.ch/record/141022/files/pdm.pdf, [Accessed: 07-Jul-2021].

[13] *Snort.org*. [Online]. Available: https://www.snort.org/downloads/community/community-rules.tar.gz, [Accessed: 07-Jul-2021].

[14] A. W. Moore and K. Papagiannaki, "'Toward the accurate identification of network applications,'" 2005, pp. 41–54.

[15] M. L. Raw Network Traffic Detection Michael J. De Lucia1, P. Maxwell2, and N. D. Bastian2, *Ananthram Swami1, Brian Jalaian1*. Nandi Lesli.

[16] F. Tegeler, X. Fu, G. Vigna, and C. Kruegel, "'Botfinder: finding bots in network traffic without deep packet inspection,'' conference on emerging networking experiments and Technologies, ser," *CoNEXT*, vol. 12, pp. 349–360, 2012.

[17] "'Malware detection by analyzing network traffic with neural networks,'" 2017, pp. 205–210.

[18] "Specification for DNS over transport layer security (TLS)," *Isi.edu*. [Online]. Available: http://www.isi.edu/%7ejohnh/PAPERS/Hu16a.html. [Accessed: 07-Jul-2021].

[19] J. Lokoc, J. Kohout, P. Cech, T. Skopal, and T. Pevný, "'k-nn classification of malware in HTTPS traffic using the metric space approach,'" 2016, pp. 131–145.

[20] B. Anderson and D. A. McGrew, "'Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity,'" 2017, pp. 1723–1732.

[21] R. Sommer and V. Paxson, "'Outside the closed world: On using machine learning for network intrusion detection,'" 2010, pp. 305–316.

[22] B. Anderson and D. A. McGrew, "'Identifying encrypted malware traffic with contextual flow data,'" 2016, pp. 35–46.